

PROIECT BAZE DE DATE

- Nume: Stoica Elias - Valeriu

- Anul 1, Semestrul 2

- Facultatea de Matematică și Informatică București

1. Descrierea modelului real, a utilitatii acestuia si a regulilor de functionare

Liga 1 de fotbal din Romania pune la start echipe din diferite colturi ale tarii. Fiecare echipa are un stadion in orasul de provenienta, unde disputa meciurile de pe propriul teren. Stadioanele se clasifica dupa capacitate, dupa numarul de stele oferit de UEFA, depinzand de cat de modern este. O echipa este alcatuita dintr-un lot de jucatori. Acestia sunt pregatiti de un antrenor principal.

Fiecare echipa are in spate un numar mare de angajati. Acestia primesc un salariu atractiv, in functie de job si de vechimea lor in club. Pentru fiecare membru din club se cunoaste tara de provenienta.

Jucatorii sunt sponsorizati de anumite companii. Acestia au un contract bine stabilit si nu pot avea mai multe sponsorizari in acelasi timp. Sponsorizarile constau in produse de marca, adidas, echipament etc. In urma acestui contract, jucatorii castiga o suma de bani, pe o anumita perioada de timp.

Campionatul de fotbal are mai multe contracte de livestreaming cu diferite televiziuni. Acestea sunt facute pe o perioada bine definita.

La finalul sezonului, echipele isi afla locul in clasament. In functie de cate puncte acumuleaza in timpul sezonului, se pot bate pentru competiile europene.

2. Prezentarea constrangerilor (restrictii, reguli) impuse asupra modelului.

Un campionat trebuie sa aiba la start mai multe echipe.

Fiecare echipa are propriul stadion.

Un jucator poate fi legitimat la o singura echipa.

Echipele au un singur antrenor principal, iar antrenorul nu poate pregati in acelasi timp mai multe echipe diferite.

Un angajat poate avea un singur job, iar la un job pot participa mai multi angajati.

Campionatul poate fi transmis la mai multe televiziuni.

Campionatul are un anumit clasament care contine echipele in functie de punctajul echipelor.

Un jucator are un contract cu un sponsor. Un sponsor atribuie un contract unui jucator. Un contract este oferit de sponsor unui jucator.

3. Descrierea entitatilor (incluzand precizarea cheilor primare)

ENTITATE	CHEIE PRIMARA	OBSERVATII
League	LeagueID	informatii generale despre ligi, precum nume, locatie sau numar de echipe.
Team	TeamID	se refera la echipele care sunt inscrise in campionatul respectiv.
Ranking	RankingID	reprezinta clasamentul ligii, iar cheia primara este chiar locul pe care se afla. de ex: RankingID = 6 ⇔ Locul al 6-lea in liga.
Broadcaster	BroadcastID	reprezinta canalele de televiziune unde se transmit meciurile in direct, acestea avand un contract stabilit pe o anumita perioada de timp.
Employee	EmployeeID	staff-ul este alcatuit din persoane care lucreaza in interiorul clubului.
Stadium	StadiumID	contine detalii despre locatia in care isi desfasoara meciurile de acasa o echipa de fotbal.
Players	PlayerID	nucleul de jucatori care evolueaza pentru o echipa.
Manager	ManagerID	antrenorul echipei, care este unic, si nu poate antrena mai multe echipe in acelasi timp
Country	CountryID	reprezinta tarile de provenienta ale jucatorilor, staff-ului, antrenorului, dar se

		leaga si de locatia stadionului, a echipei de fotbal.
Job	JobID	informatii despre fiecare job din cadrul clubului, avand angajati pe cei din tabela 'Staff'
Sponsor	SponsorID	informatii despre sponsorii care pot reprezenta jucatorii
Contract	ContractID	informatii despre un contract intre jucator si sponsor
Livestream	LeagueID, BroadcastID	tabela intermediara, care descrie o relatie de tip many to many.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

RELATIE	CARDINALITATE	OBSERVATII
Is live on	Liga1 - Broadcasting(one to many)	Liga poate avea mai multe contracte de televizare, iar un canal TV televizeaza doar liga noastra.
Deal	Players - Contract - Sponsor (one to one) Relatie ternara	Un jucator are un contract cu un anumit sponsor. Un contract este realizat intre un jucator si un sponsor. Un sponsor realizeaza un contract cu un jucator.
Is trained by	Teams - Manager(one to one)	O echipa poate fi antrenata de un antrenor unic.
Work at	Employees - Jobs(many to one)	Un angajat poate avea un singur job. Un job poate fi practicat de mai multi angajati.
Is located in	Stadium - Cities(one to one)	Un stadion se afla intr-un oras.
Consists of	Liga1 - Teams(one to many)	Liga are mai multe echipe la start.

Plays On	Teams -Stadium (one to one)	Fiecare echipa are stadionul ei, unde isi joace meciurile. Un stadion apartine unei singure echipe.
----------	-----------------------------	---

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

Entitate: League

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
LeagueName	string			NOT NULL
LeagueLocation	string			NOT NULL

Entitate: Team

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
TeamName				
FoundationYear				

Entitate: Ranking

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
Points				

Entitate: Broadcaster

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
BroadcasterName				

CommentatorName				
StartDate				
EndDate				

Entitate: Employee

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatori
EmployeeName				

Entitate: Stadium

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatori
StadiumName				
StadiumCapacity				
BuildingYear				
UefaStars				
StadiumCity				

Entitate: Player

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatori
PlayerName				
PlayerPosition				
PlayerJerseyNo				

Entitate: Manager

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
ManagerName				

Entitate: Country

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
CountryName				
CountryContinent				

Entitate: Job

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
JobName				
JobSalary				

Entitate: Sponsor

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
SponsorName				

Entitate: Contract

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
Value				

Years				
-------	--	--	--	--

Relatie: Livestream

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
LeagueID				
BroadcasterID				

Relatie: Work at

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
Employees				
Jobs				

Relatie: Consists of

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
League				
Teams				

Relatie: Plays on

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatoriu
Teams				
Stadium				

Relatie: Is trained by

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatorio
Teams				
Manager				

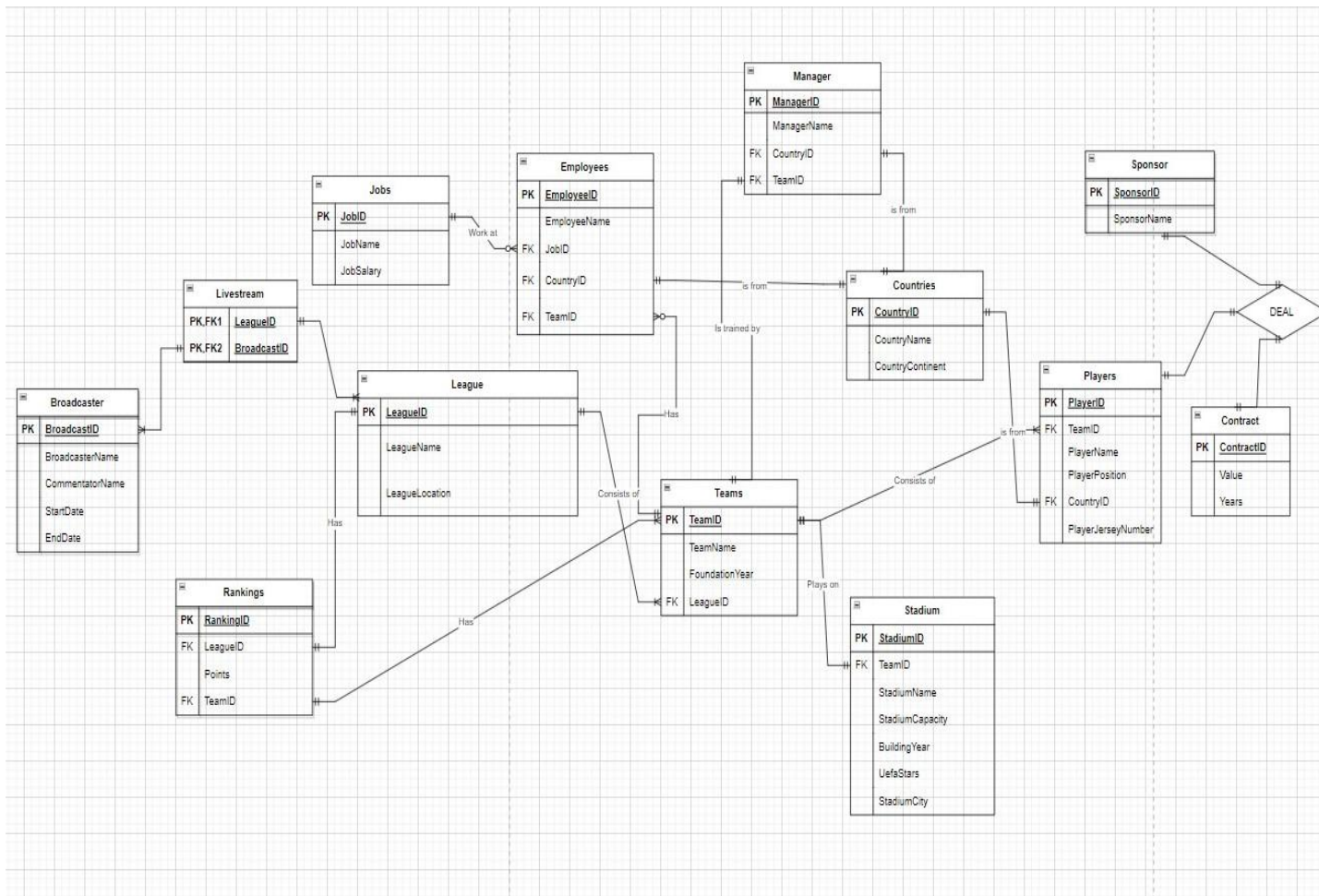
Relatie: Deal

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatorio
Players				
Sponsor				
Contract				

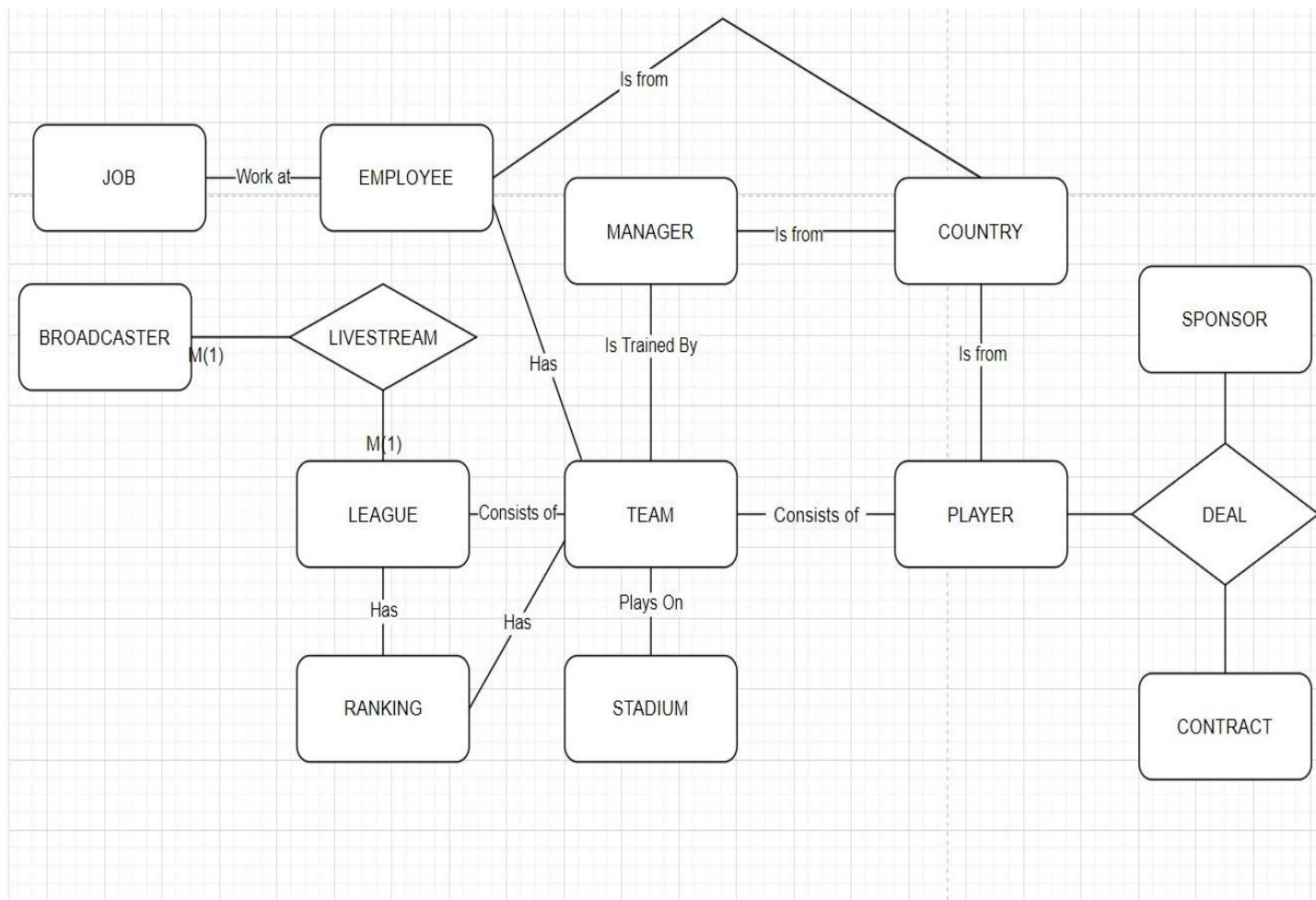
Relatie: Is located in

Atribut	Tip	Dimensiune	Valori posibile si valori default	Observatii/ Obligatorio
Stadium				
Cities				

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzatoare diagramei entitate-relatie proiectate la punctul 6.



8. Enumerarea schemelor relationale corespunzatoare diagramei conceptuale proiectate la punctul 7

LEAGUE(LeagueID#, LeagueName, LeagueLocation)
 BROADCASTER(BroadcastID#, BroadcasterName, StartDate, EndDate)
 LIVESTREAM(LeagueID#, BroadcastID#)
 RANKING(RankingID#, LeagueID(FK), TeamID(FK), Points)
 TEAM(TeamID#, LeagueID(FK), TeamName, FoundationYear)
 STADIUM(StadiumID#, TeamID(FK), StadiumName, StadiumCapacity, BuildingYear, UefaStars, StadiumCity, StadiumAdress)
 COUNTRIES(CountryID#, CountryName, CountryContinent)
 JOBS(JobID#, JobName, JobSalary)
 EMPLOYEES(EmployeeID#, JobID(FK), CountryID(FK), TeamID(FK), EmployeeName)

MANAGER(ManagerID#, ManagerName, TeamID(FK), CountryID(FK))
 PLAYER(PlayerID#, TeamID(FK), CountryID(FK), PlayerName, PlayerPosition, PlayerJerseyNumber)
 SPONSOR(SponsorID#, SponsorName)
 CONTRACT(ContractID#, Value, Years)
 DEAL(PlayerID#, SponsorID#, ContractID#)

9. Realizarea normalizarii pana la FN3 (FN1-FN3)

FN1: Identificam attributele care erau formate din attribute atomice si le separam. Baza de date este in FN1 deja, deci un exemplu negativ ar fi:

Non-FN1:

BroadcasterID	LeagueID
1	1, 2, 3
2	2, 4
3	4
4	3

FN1:

BroadcasterID	LeagueID
1	1
1	2
1	3
2	2
2	4
3	4
4	3

FN2: Baza de date este deja in FN2, deoarece nu intalnim nicio cheie primara compusa in nicio entitate.

FN3: Baza de date este in FN3. In cadrul acestei baze de date, nu exista cazul ca mai multe attribute sa nu depinda de cheia primara. Un exemplu de NON-FN3 ar fi: StadiumCity si StadiumAdress depind doar de orasul in care se afla stadionul, nu de Stadion. Putem separa attributele intr-o tabela City(CityID#, CityName), Address(AdressID#, AdressStreet, CityID(FK))

NONFN3:

STADIUM	STADIUMCITY	STADIUMADRESS
Arena Nationala	Bucuresti	Bulevardul Pierre Coubertin
Stefan Cel Mare	Bucuresti	Soseaua Colentina
Stadionul Otelul	Galati	Strada Siderugistului

FN3

STADIUM	ADRESSID
Arena Nationala	1
Stefan cel Mare	2
Stadionul Otelul	3

CITYID	CITYNAME
1	BUCURESTI
2	GALATI

ADRESSID	ADRESS_STREET	CITYID
1	Soseaua Colentina	1
2	Bulevardul Pierre Coubertin	1
3	Strada Siderugistului	2

BCNF: O baza de date este in BCNF daca si numai daca nu exista nicio dependenta functionala de atribute non-triviale pe orice altceva diferit de un superset de chei ale unei chei de candidat.

Un caz ipotetic de NON-BNCF ar fi:

(LeagueID, BroadcasterName) -> CommentatorName

(LeagueID, CommentatorName) -> BroadcasterName

BroadcasterName -> CommentatorName

NON-BCNF:

LeagueID	BroadcasterName	CommentatorName
1	DigiSport	Dan Stefanescu
1	ProTV	Mihai Mironica
2	TVR	Marian Oleianos
3	LookTV	Emil Gradinescu
4	DigiSport	Dan Stefanescu

BCNF:

R1

BroadcasterID	BroadcasterName	CommentatorName
1	DigiSport	Dan Stefanescu
2	ProTV	Mihai Mironica
3	TVR	Marian Oleianos
4	LookTV	Emil Gradinescu

R2

LeagueID	BroadcasterID
1	1
1	2
2	3
3	4
4	1

10. Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea (minimum 5 inregistrari in fiecare tabel neasociativ, minimum 10 inregistrari in tabelele asociative)

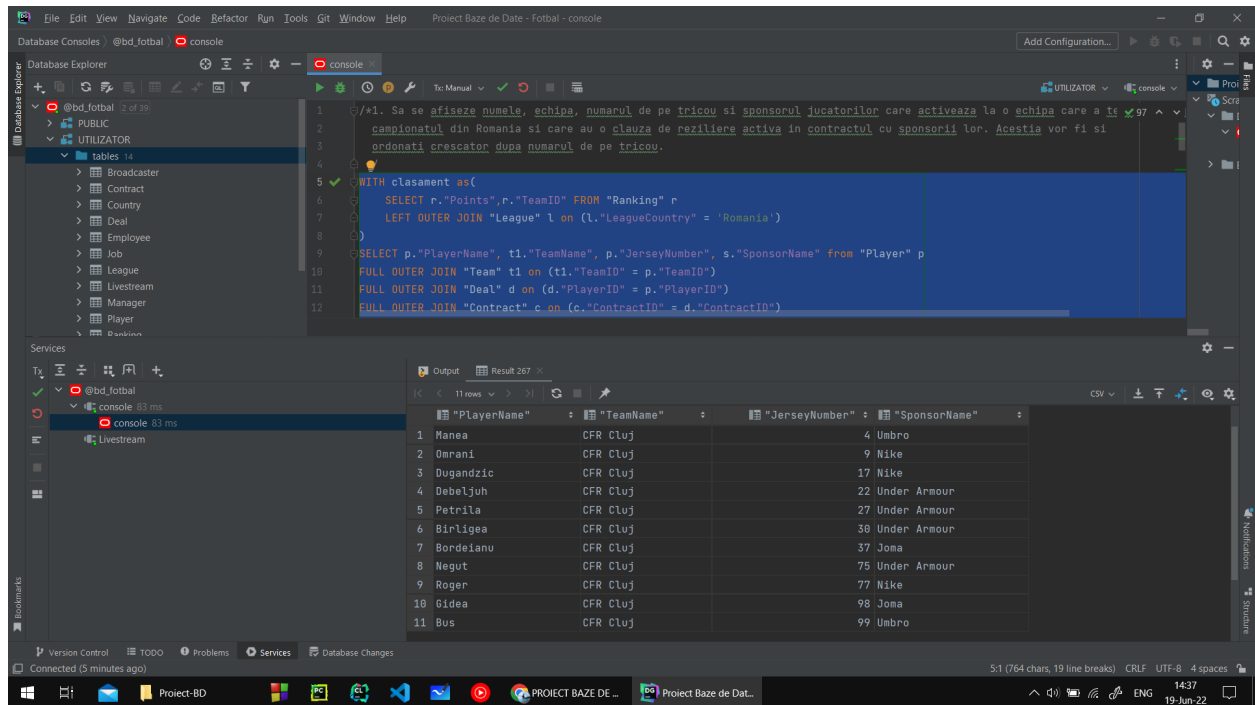
Fisierele de creare si inserare sunt intr-un repository pe github, la adresa urmatoare:

<https://github.com/wood11nho/Proiect-BD>

11. Formulati in limbaj natural si implementati 5 cereri SQL complexe ce vor utiliza, in ansamblul lor anumite elemente.

/*1. Sa se afiseze numele, echipa, numarul de pe tricou si sponsorul jucatorilor care activeaza la o echipa care a terminat pe primul loc in campionatul din Romania si care au o clauza de reziliere activa in contractul cu sponsorii lor. Acestia vor fi si ordonati crescator dupa numarul de pe tricou.

```
*/
WITH clasament as(
    SELECT r."Points",r."TeamID" FROM "Ranking" r
    LEFT OUTER JOIN "League" l on (l."LeagueCountry" = 'Romania')
)
SELECT p."PlayerName", t1."TeamName", p."JerseyNumber", s."SponsorName" from
"Player" p
FULL OUTER JOIN "Team" t1 on (t1."TeamID" = p."TeamID")
FULL OUTER JOIN "Deal" d on (d."PlayerID" = p."PlayerID")
FULL OUTER JOIN "Contract" c on (c."ContractID" = d."ContractID")
FULL OUTER JOIN "Sponsor" s on (s."SponsorID" = d."SponsorID")
WHERE t1."TeamID" =
(
    SELECT t."TeamID" from "Team" t
    FULL OUTER JOIN clasament c on (t."TeamID" = c."TeamID")
    WHERE c."Points" = (
        SELECT MAX(c1."Points") FROM clasament c1
    )
)
AND
c."ClauzaReziliere" = 1
ORDER BY p."JerseyNumber";
```



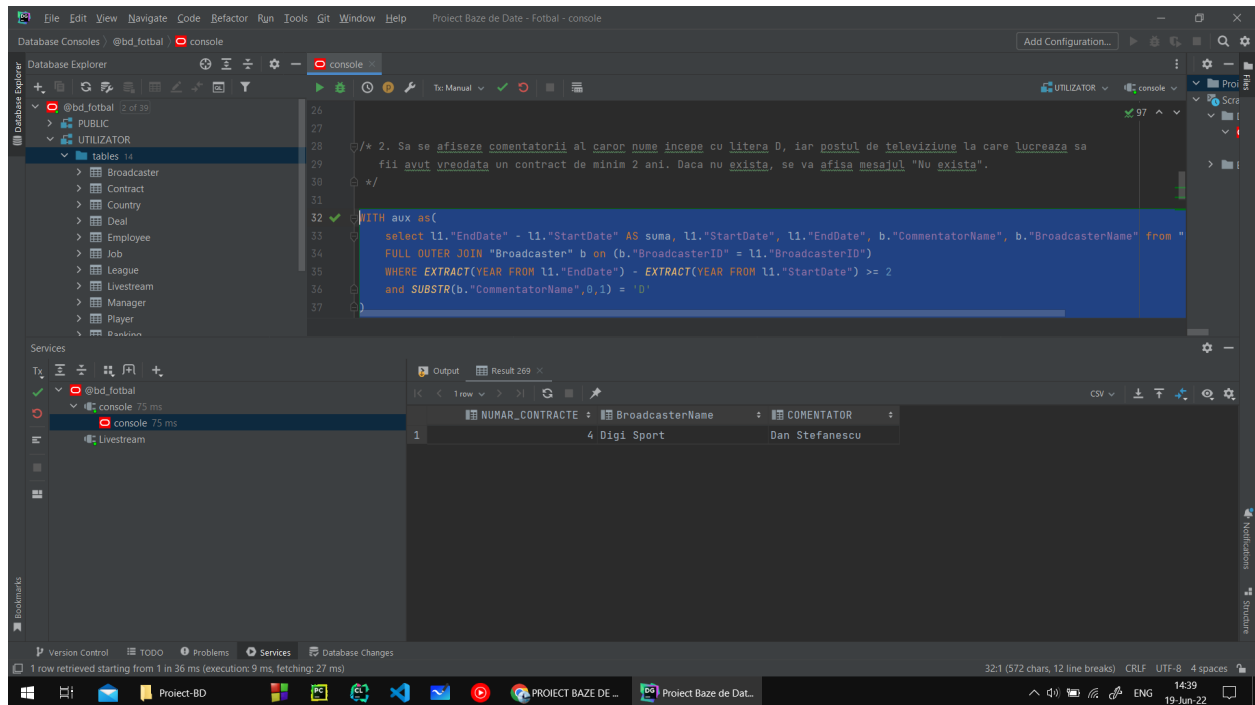
/* 2. Sa se afiseze comentatorii al caror nume incepe cu litera D, iar postul de televiziune la care lucreaza sa fii avut vreodata un contract de minim 2 ani. Daca nu exista, se va afisa mesajul "Nu exista".

***/**

```

WITH aux as(
    select l1."EndDate" - l1."StartDate" AS suma, l1."StartDate", l1."EndDate",
    b."CommentatorName" from "Livestream" l1
    FULL OUTER JOIN "Broadcaster" b on (b."BroadcasterID" = l1."BroadcasterID")
    WHERE EXTRACT(YEAR FROM l1."EndDate") - EXTRACT(YEAR FROM
l1."StartDate") >= 2
    and SUBSTR(b."CommentatorName",0,1) = 'D'
)
SELECT COUNT(suma) as Numar_Contracte,
CASE
    WHEN COUNT(suma) >= 1 THEN "CommentatorName"
    ELSE 'Nu exista'
END AS Comentator
FROM aux
group by "CommentatorName";

```

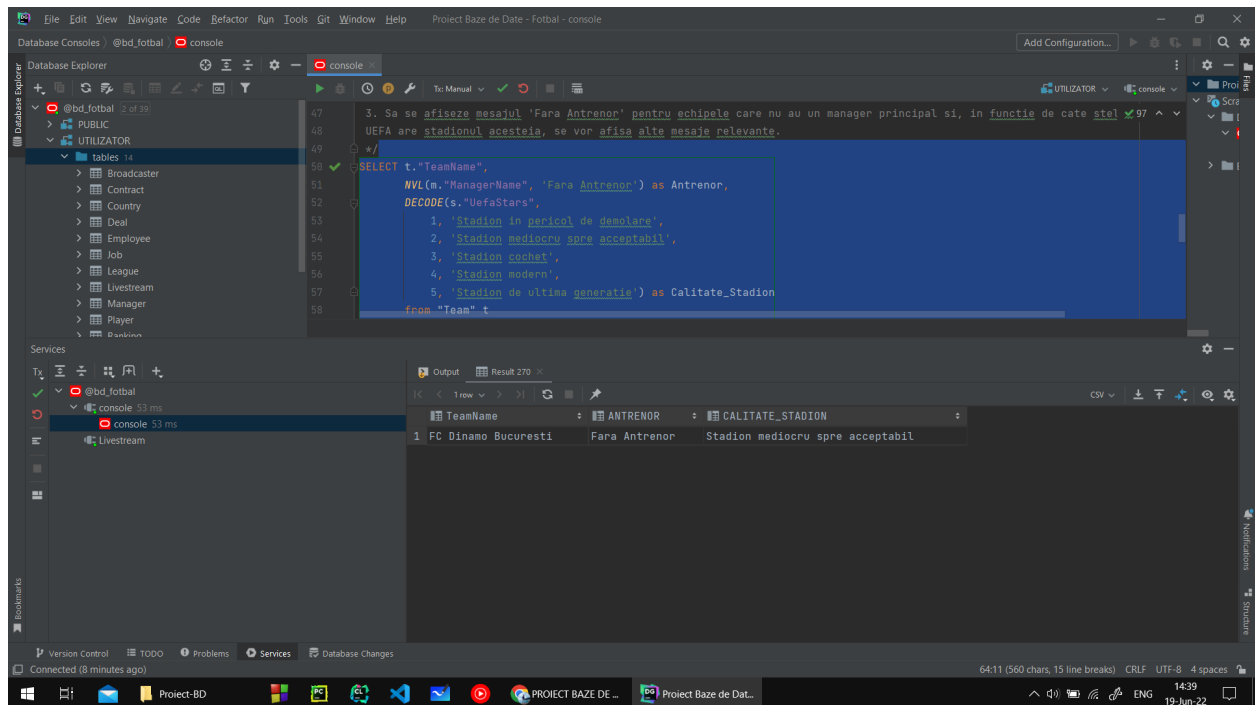


/*

3. Sa se afiseze mesajul 'Fara Antrenor' pentru echipele care nu au un manager principal si, in functie de cate stele UEFA are stadionul acestuia, se vor afisa alte mesaje relevante.

*/

```
SELECT t."TeamName",
       NVL(m."ManagerName", 'Fara Antrenor') as Antrenor,
       DECODE(s."UefaStars",
              1, 'Stadion in pericol de demolare',
              2, 'Stadion mediocru spre acceptabil',
              3, 'Stadion cochet',
              4, 'Stadion modern',
              5, 'Stadion de ultima generatie') as Calitate_Stadion
from "Team" t
FULL OUTER JOIN "Manager" m on (m."TeamID" = t."TeamID")
FULL OUTER JOIN "Stadium" s on (s."TeamID" = t."TeamID")
where t."TeamID" not in
(
  SELECT m1."TeamID" FROM "Manager" m1
);
```

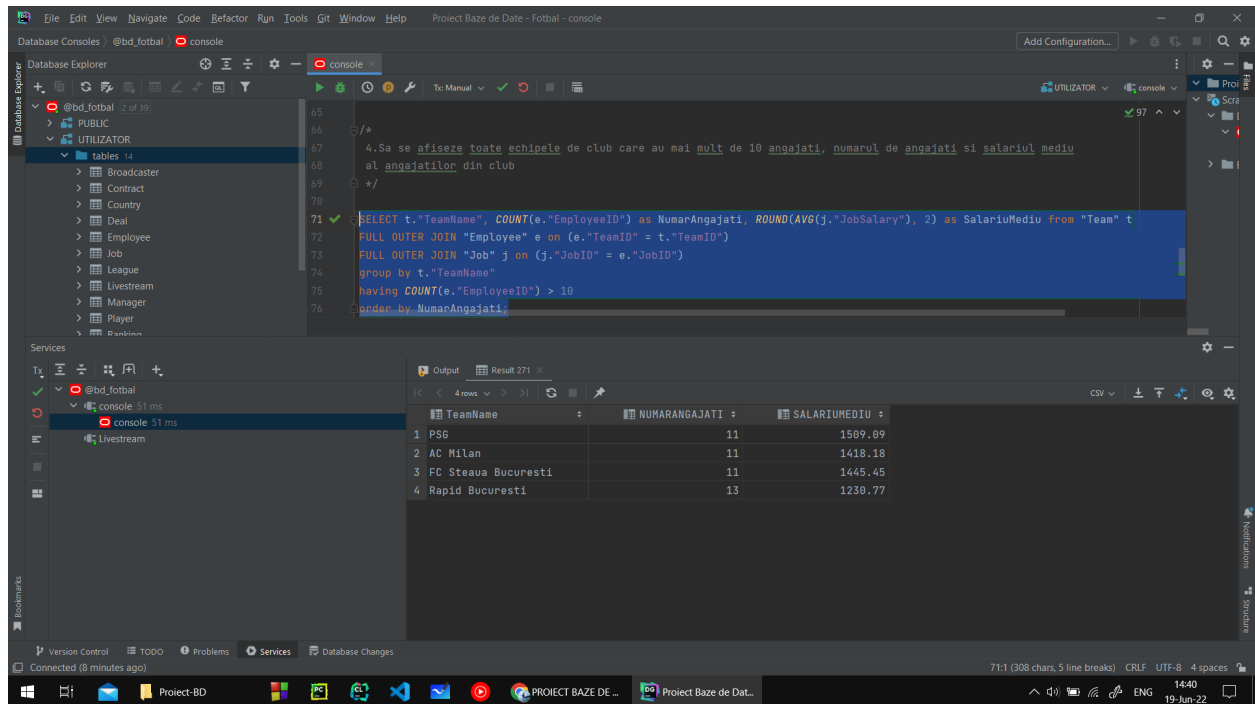



/*

4.Sa se afiseze toate echipele de club care au mai mult de 10 angajati, numarul de angajati si salariul mediu al angajatilor din club
***/**

```

SELECT      t."TeamName",      COUNT(e."EmployeeID")      as      NumarAngajati,
ROUND(AVG(j."JobSalary"), 2) as SalariuMediu from "Team" t
FULL OUTER JOIN "Employee" e on (e."TeamID" = t."TeamID")
FULL OUTER JOIN "Job" j on (j."JobID" = e."JobID")
group by t."TeamName"
having COUNT(e."EmployeeID") > 10
order by NumarAngajati;
  
```



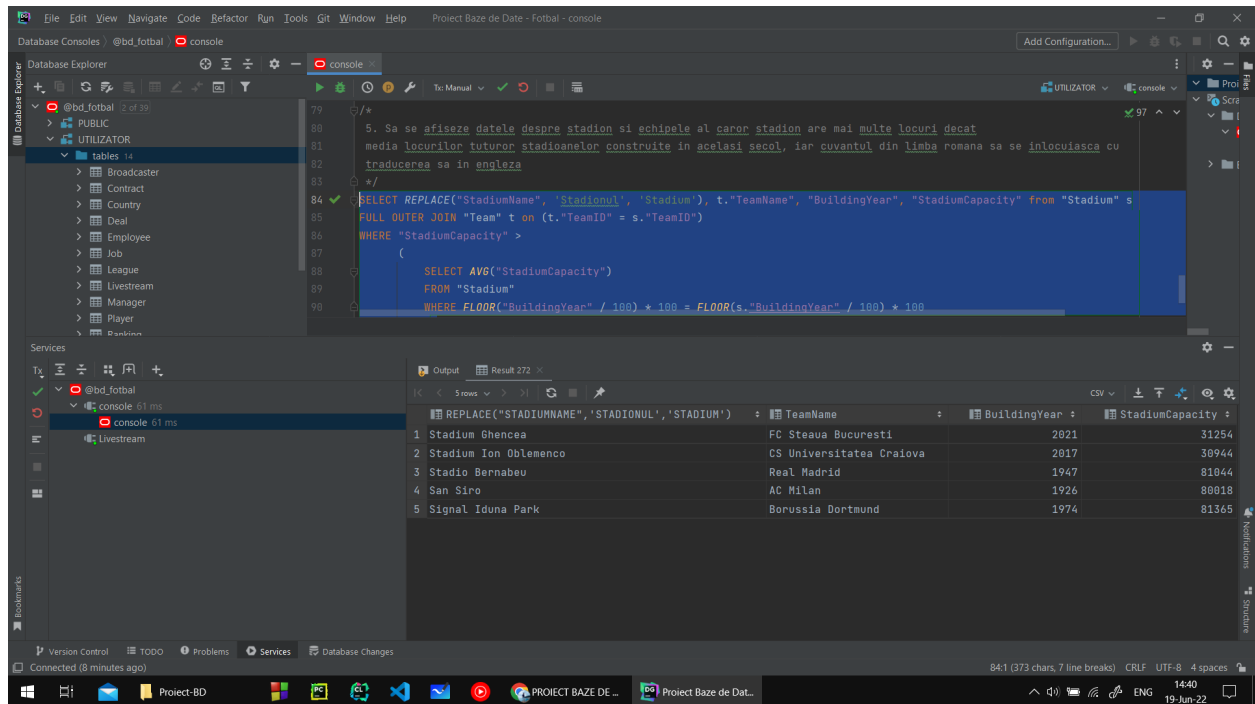
/* 5. Sa se afiseze datele despre stadion si echipele al caror stadion are mai multe locuri decat media locurilor tuturor stadioanelor construite in acelasi secol, iar cuvantul din limba romana sa se inlocuiasca cu traducerea sa in engleza.

***/**

```

SELECT REPLACE("StadiumName", 'Stadionul', 'Stadium'), t."TeamName",
"BuildingYear", "StadiumCapacity" from "Stadium" s
FULL OUTER JOIN "Team" t on (t."TeamID" = s."TeamID")
WHERE "StadiumCapacity" >
(
    SELECT AVG("StadiumCapacity")
    FROM "Stadium"
    WHERE FLOOR("BuildingYear" / 100) * 100 = FLOOR(s."BuildingYear" / 100) *
100
);

```



12. Sa se implementeze minim 3 operatii de actualizare sau de suprimare a datelor, utilizand subcereri.

/*

1. Sa se modernizeze stadioanele care s-au construit mai devreme de anul 1950, cu o capacitate cu 10% mai mare decat cea a celui mai mare stadion si cu 5 stele UEFA

*/

```

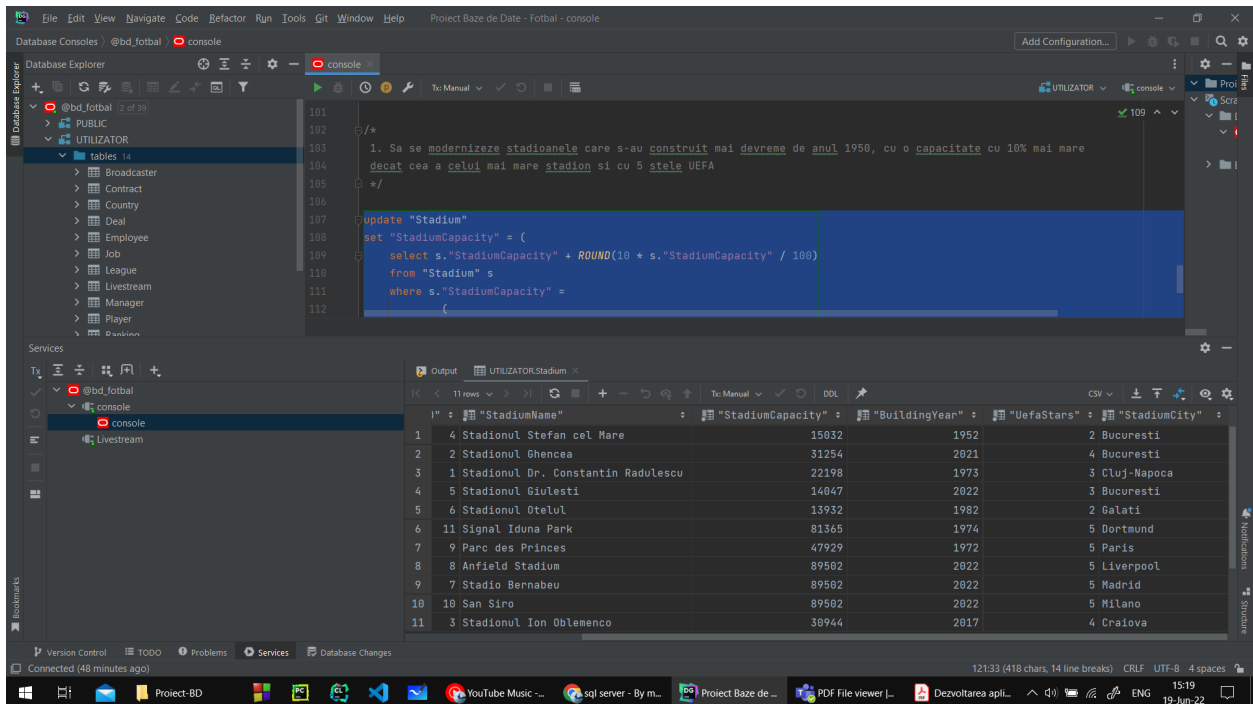
update "Stadium"
set "StadiumCapacity" = (
    select s."StadiumCapacity" + ROUND(10 * s."StadiumCapacity" / 100)
    from "Stadium" s
    where s."StadiumCapacity" =
        (
            select MAX("StadiumCapacity")
            from "Stadium"
        )
),
"UefaStars" = 5,
"BuildingYear" = (

```

```

select EXTRACT(Year from CURRENT_DATE) FROM DUAL
)
where "BuildingYear" < 1950;

```



```

/*

```

2. Sa se produca o modificare in tabel astfel incat contractele de sponsorizare care mai dureaza un singur an sa expire. (sa fie sterse)

```

*/

```

```

DELETE FROM "Deal" d
WHERE d."ContractID" in
(
    select "ContractID" from "Contract"
    where "Years" = 1
);

```

```

select * from "Deal"
where "ContractID" = 1 or "ContractID" = 6

```

INAINTE DE DELETE:

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure for 'bd_fotbal', including tables like 'Broadcast', 'Contract', 'Country', 'Deal', 'Employee', 'Job', 'League', 'Livestream', 'Manager', 'Player', 'Ranking', 'Sponsor', 'Stadium', and 'Team'. The right pane shows a SQL script in the 'console' window. The script includes a comment in Romanian: '2. Sa se produca o modificare in tabel astfel incat contractele de sponsorizare care mai dureaza un singur an sa expire. (sa fie sterse)'. The SQL code is as follows:

```
/*
2. Sa se produca o modificare in tabel astfel incat contractele de sponsorizare care mai dureaza un singur an
sa expire. (sa fie sterse)
*/
DELETE FROM "Deal" d
WHERE d."ContractID" in
(
    select "ContractID" from "Contract"
    where "Years" = 1
);
select * from "Deal"
where "ContractID" = 1 or "ContractID" = 6
```

The 'Output' window shows the results of the query, displaying 33 rows of data with columns 'PlayerID', 'SponsorID', and 'ContractID'.

PlayerID	SponsorID	ContractID
1	2	1
2	6	3
3	12	4
4	21	3
5	30	1
6	36	5
7	37	6

DUPA DELETE:

The screenshot shows the same SQL Server Enterprise Manager interface as before, but the 'Output' window now shows 0 rows of data, indicating that the deletion was successful. The SQL script in the 'console' window is the same as in the previous screenshot.

```
/*
2. Sa se produca o modificare in tabel astfel incat contractele de sponsorizare care mai dureaza un singur an
sa expire. (sa fie sterse)
*/
DELETE FROM "Deal" d
WHERE d."ContractID" in
(
    select "ContractID" from "Contract"
    where "Years" = 1
);
select * from "Deal"
where "ContractID" = 1 or "ContractID" = 6
```

The 'Output' window shows 0 rows of data, confirming that the deletion was successful.

/*

3. Sa se mareasca cu 20% salariile din cadrul jobului/joburilor cele mai prost platite

*/

```
UPDATE "Job"
SET "JobSalary" =
(
    "JobSalary" * 1.2
)
where "JobSalary" = (
    select MIN("JobSalary") FROM "Job"
);

SELECT * FROM "Job";
```

INAINTE:

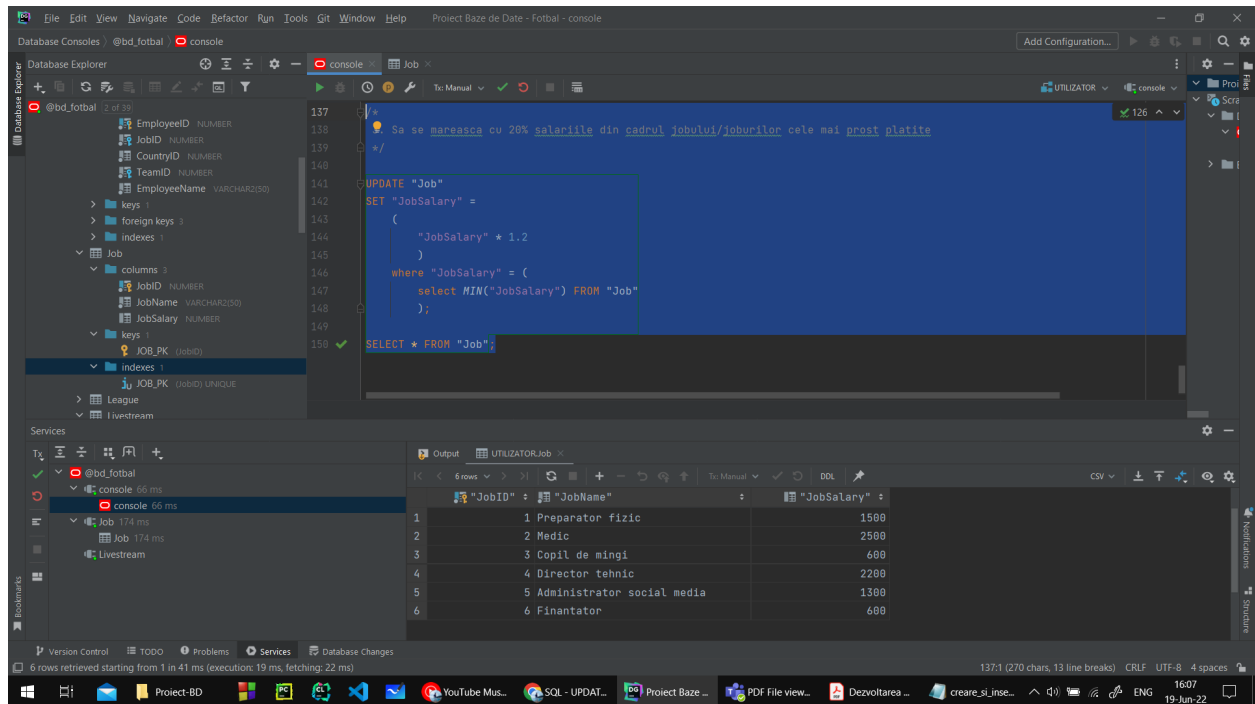
The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure for 'bd_fotbal', including tables like 'Employee', 'Job', and 'Livestream'. The right pane shows the execution of a query in the 'console' window. The query is a multi-statement batch:

```
/*
3. Sa se mareasca cu 20% salariile din cadrul jobului/joburilor cele mai prost platite
*/
UPDATE "Job"
SET "JobSalary" =
(
    "JobSalary" * 1.2
)
where "JobSalary" = (
    select MIN("JobSalary") FROM "Job"
);
SELECT * FROM "Job";
```

The output window shows the results of the query, displaying a table with 6 rows and 3 columns: 'JobID', 'JobName', and 'JobSalary'.

JobID	JobName	JobSalary
1	Preparator fizic	1500
2	Medic	2500
3	Copil de mingi	500
4	Director tehnic	2200
5	Administrator social media	1300
6	Finantator	600

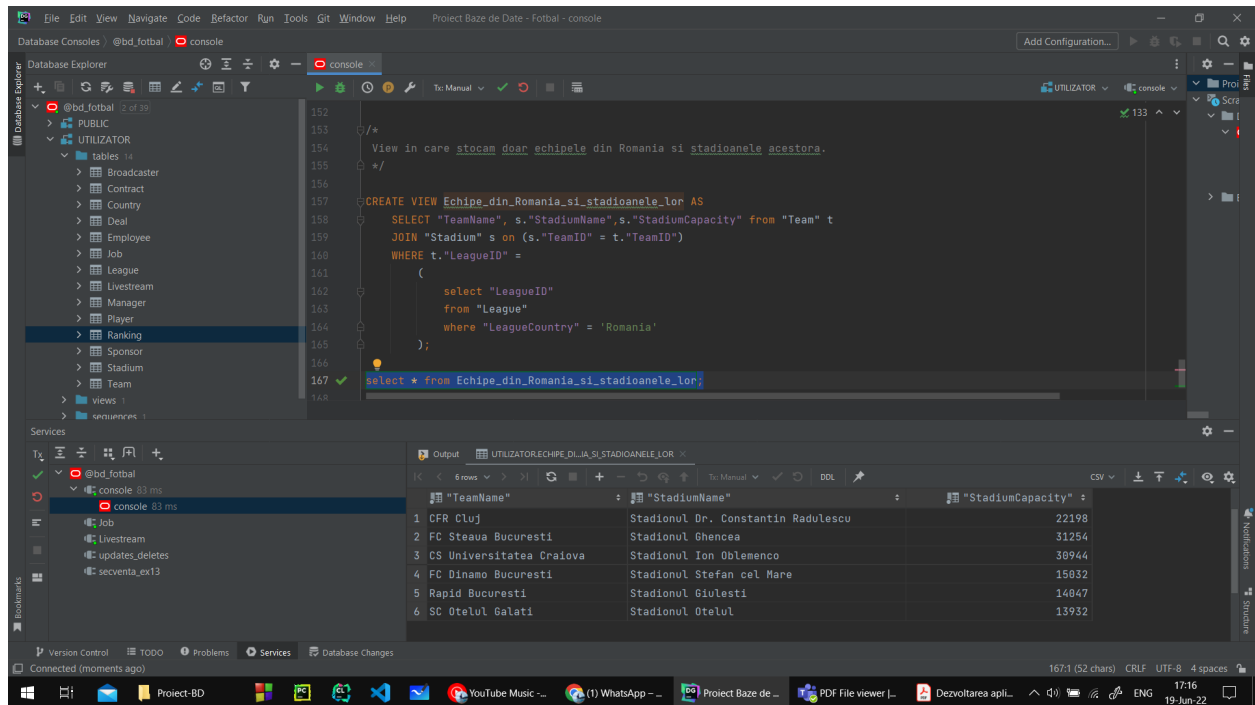
DUPA:



13. Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in table (punctul 10)



14. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.



```
CREATE VIEW Echipe_din_Romania_si_stadioanele_lor AS  
    SELECT "TeamName", s."StadiumName",s."StadiumCapacity" from  
    "Team" t  
    JOIN "Stadium" s on (s."TeamID" = t."TeamID")  
    WHERE t."LeagueID" =  
    (  
        select "LeagueID"  
        from "League"  
        where "LeagueCountry" = 'Romania'  
    );
```


Operatie LMD permisa: modificarea unui singur tabel

```
CREATE VIEW Echipe_din_Romania_si_stadioanele_lor AS
SELECT "TeamName", s."StadiumName",s."StadiumCapacity" from "Team" t
JOIN "Stadium" s on (s."TeamID" = t."TeamID")
WHERE t."LeagueID" =
(
select "LeagueID"
from "League"
where "LeagueCountry" = 'Romania'
);

select * from Echipe_din_Romania_si_stadioanele_lor;

UPDATE Echipe_din_Romania_si_stadioanele_lor SET "StadiumName" = REPLACE("StadiumName",'Stadionul', 'Stadium')
```

Output UTILIZATOR.ECHIBE_DI...IA_SI_STADIOANELE_LOR

	"TeamName"	"StadiumName"	"StadiumCapacity"
1	CFR Cluj	Stadium Dr. Constantin Radulescu	22198
2	FC Steaua Bucuresti	Stadium Ghencea	31254
3	CS Universitatea Craiova	Stadium Ion Oblemenco	30944
4	FC Dinamo Bucuresti	Stadium Stefan cel Mare	15032
5	Rapid Bucuresti	Stadium Giulesti	14047
6	SC Otelul Galati	Stadium Otelul	13932

Operatie LDM nepermisa: modificarea ambelor tabele din acest view

```
158 SELECT "TeamName", s."StadiumName",s."StadiumCapacity" from "Team" t
159 JOIN "Stadium" s on (s."TeamID" = t."TeamID")
160 WHERE t."LeagueID" =
161 (
162 select "LeagueID"
163 from "League"
164 where "LeagueCountry" = 'Romania'
165 );
166
167 select * from Echipe_din_Romania_si_stadioanele_lor;
168
169 UPDATE Echipe_din_Romania_si_stadioanele_lor SET "TeamName" = UPPER("TeamName"),
170 "StadiumName" = REPLACE("StadiumName",'Stadium', 'Stadionul');
```

[42000][1776] ORA-01776: cannot modify more than one base table through a join view
Position: 131

Output UTILIZATOR.ECHIBE_DI...IA_SI_STADIOANELE_LOR

	"TeamName"	"StadiumName"	"StadiumCapacity"
1	CFR Cluj	Stadium Dr. Constantin Radulescu	22198
2	FC Steaua Bucuresti	Stadium Ghencea	31254
3	CS Universitatea Craiova	Stadium Ion Oblemenco	30944
4	FC Dinamo Bucuresti	Stadium Stefan cel Mare	15032
5	Rapid Bucuresti	Stadium Giulesti	14047
6	SC Otelul Galati	Stadium Otelul	13932

15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.
16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.

/*1. Sa se afiseze numele, echipa, numarul de pe tricou si sponsorul jucatorilor care activeaza la o echipa care a terminat pe primul loc in campionatul din Romania si care au o clauza de reziliere activa in contractul cu sponsorii lor. Acestia vor fi si ordonati crescator dupa numarul de pe tricou.

```

*/
WITH clasament as(
    SELECT r."Points",r."TeamID" FROM "Ranking" r
    LEFT OUTER JOIN "League" l on (l."LeagueCountry" = 'Romania')
)
SELECT p."PlayerName", t1."TeamName", p."JerseyNumber", s."SponsorName" from
"Player" p
FULL OUTER JOIN "Team" t1 on (t1."TeamID" = p."TeamID")
FULL OUTER JOIN "Deal" d on (d."PlayerID" = p."PlayerID")
FULL OUTER JOIN "Contract" c on (c."ContractID" = d."ContractID")
FULL OUTER JOIN "Sponsor" s on (s."SponsorID" = d."SponsorID")
WHERE t1."TeamID" =
(
    SELECT t."TeamID" from "Team" t
    FULL OUTER JOIN clasament c on (t."TeamID" = c."TeamID")
    WHERE c."Points" = (
        SELECT MAX(c1."Points") FROM clasament c1
    )
)
AND
    c."ClauzaReziliere" = 1
ORDER BY p."JerseyNumber";

```

/*

Division

1. Sa se obtina numele sponsorilor care au contracte cu toti jucatorii care evolueaza in poarta.

*/

```
SELECT s."SponsorName" FROM "Deal" d
JOIN "Sponsor" s on (s."SponsorID" = d."SponsorID")
WHERE d."PlayerID" in
(
    SELECT "PlayerID"
    FROM "Player"
    WHERE "PlayerPosition" = 'GK'
)
GROUP BY s."SponsorName"
HAVING COUNT(d."PlayerID") = (
    SELECT COUNT(*)
    FROM "Player"
    WHERE "PlayerPosition" = 'GK'
)
```

/*

2. Sa se obtina numele ligilor care sunt transmise pe toate canalele care contin 'Sport' in nume

*/

```
SELECT b."BroadcasterName" FROM "Livestream" l
JOIN "Broadcaster" b on (b."BroadcasterID" = l."BroadcasterID")
WHERE l."BroadcasterID" in
(
    SELECT "BroadcasterID"
    FROM "Broadcaster"
    WHERE "BroadcasterName" LIKE '%Sport%'
)
GROUP BY b."BroadcasterName"
HAVING COUNT(l."BroadcasterID") = (
    SELECT COUNT(*)
    FROM "Broadcaster"
    WHERE "BroadcasterName" LIKE '%Sport%'
)
```

